

Virtual Domotic Systems: a 3D Interaction Technique to Control Virtual Building Devices Using Residential Gateways

Francisco Pedro Luque¹, Francisco Javier Luque¹, Carmen Lastres¹, Asunción Santamaría¹, Iris Galloso¹, Jordi Muria²

¹CeDint - Universidad Politécnica de Madrid
Campus de Montegancedo, 28223 Pozuelo de Alarcón - Madrid (Spain)

e-mail: [franluque, fluque, clastres, asun,iris]@cedint.upm.es

²T-Systems Iberia
e-mail: jordi.muria@t-systems.es

Abstract— Residential gateways are systems based on different control technologies that provide a unified way of communication for various devices and appliances present in homes and buildings (sensors, control systems, electronic devices, etc.). The massive introduction of residential gateways in the market is often hindered by the lack of intuitive configuration and visualization interfaces. Moreover, users frequently obtain very limited feedback about the status of the building after the actions carried out by the residential gateway, what reduces the confidence in such systems.

This paper presents a Virtual Domotic System (VDS), an innovative solution to provide a Virtual Reality interface to manage residential gateways. VDS comprises three main blocks. The first of them is the residential gateway and the associated control devices of the building. The second element is an advanced 3D Virtual Environment that reliably represents the building, the state of control devices and the environmental characteristics (such as lighting or temperature). Finally, the system includes the software that enables the communication and synchronization between the virtual environment and the control technologies and appliances of the building.

The VDS introduces a new 3D interaction technique, where the inputs that modify and configure the 3D virtual environment come directly from the sensors and actuators installed in the building.

Keywords— User interface, 3D interaction technique, Domotic devices, Virtual Reality, Residential gateway.

I. INTRODUCTION

There is currently a wide variety of residential gateways on the market. They are systems based on different control technologies that provide a unified way of communicating and controlling domotic devices and appliances in homes or buildings. The use of this type of equipment is beneficial in terms of efficient energy management and control of complex domotic device networks [1].

Despite these benefits, potential users are reluctant to install residential gateways on buildings or even in their own homes. One of the main reasons is the non-intuitive user interfaces commonly used in this kind of systems. In most cases, interactions are based on 2D GUI menus or even text and command lines, difficult to learn and operate for unskilled users.

During the last years, new ways of interaction have been studied in this area. For example, a control application based on gaze and eye tracking is described in [2] and [3]. This paper presents the results of an R&D project that uses Virtual Reality environments to create a 3D interface for residential gateways.

The idea of using 3D environments to operate residential gateways comes from several studies, carried out to determine the effectiveness of using virtual 3D interfaces compared to common 2D GUIs. One of them has shown that using a 3D map to assist navigation is considerably more efficient in terms of time to complete tasks [4]. This effectiveness can be explained by the advantages offered by the Virtual Environments: sense of immersion, similarity to the real world and therefore, intuitive use.

The final objective of the project has been the design and development of a Virtual Domotic System (VDS) to provide a more usable and intuitive interface to interact with commercial residential gateways [5]. We consider this is of particular importance not only for consumer market in smart house controlling but also to help people with special needs such as those with mobility disabilities [2]. Fig. 1 presents the main blocks of the VDS and the connections between them.

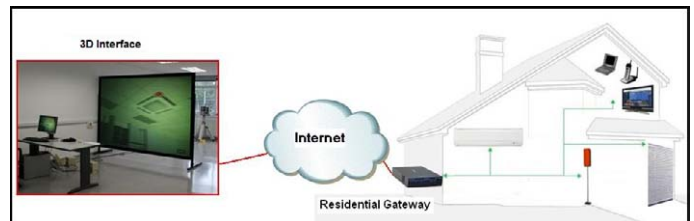


Fig. 1. Virtual domotic system. General diagram.

The diagram shows on the right side the residential gateway connected to the network of domotic devices. On the left side, the 3D application is presented, from where user interacts with the residential gateway and the domotic network. Both blocks are interconnected through internet, so the VDS does not need to be in the same place as the building.

The 3D interface is based on a virtual model that accurately reproduces the real building, including every appliance that can be controlled from the residential gateway. The navigation inside the scene is performed from a first person perspective. Experience with virtual reality navigation has demonstrated that constraining and guiding the users like they would do in the real world produces more effective navigation and orientation [4].

The devices modeled include actuators such as heating systems, dimmer lights and shutters. They are highlighted when the user points at them, so any previous knowledge on which appliances can be controlled is not needed. The model reacts consequently to the current

building state and more than one device can affect the virtual representation of the building. For example, the illumination of a room is calculated and presented proportionally to the intensity of artificial light (dimmerable lamps) and natural lighting (degree of aperture of shutters). The consequences of users' actions are shown as they would be perceived in the real world. This kind of feedback is more intuitive than any other used in current 2D interfaces.

The main contribution of the VDS is the development of a synchronization service between the real world (domotic devices) and its virtual representation (3D model). This service allows the system to communicate in both directions. In this way, actions carried out in the real world are immediately updated in the virtual representation of the building. For example, if a user enters a room and switches on a light, the virtual model shows a warning message and updates the virtual model with the current status of the lighting.

The key to achieve this interactivity is a new 3D interaction technique developed for this project. This technique uses the residential gateway to communicate with the VDS and update the virtual building according to the state of domotic devices.

Moreover, this technique makes the system capable to operate in multi-user scenarios where several users can access the residential gateway simultaneously to control either the same room or different rooms of the same building. In the other direction, multiple clients of the application can establish a connection with the same building environment and send actions to any set of home appliances in a collaborative way.

Finally, it should be highlighted that the system can be operated from all kinds of Virtual Reality sets, that range from simple non-immersive systems, such as a PC with a keyboard and a mouse, to sophisticated visualization immersive environments such as stereoscopic projection systems (stereowall, CAVE, etc.) [6], [7], [8].

II. SYSTEM ARCHITECTURE

The architecture of the virtual domotic system comprises the following elements, as shown in Fig. 2:

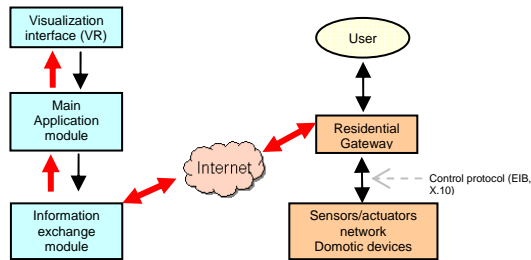


Fig. 2. Architecture of the system

- **Domotic devices.** Set of devices used to control electronic appliances present at homes or buildings. With them, users can interact with the virtual representation of the building. Two main types of domotic devices can be found:

- **Sensors.** Used to measure specific parameters (temperature, luminance, humidity, etc.).
- **Actuators.** Used to control (open, close, switch, dim, etc.) electronic devices that operate on lights, shutters, blinds, heating/air conditioning, etc.

- **Residential gateway.** Platform to provide access and control to domotic devices. They are intended to manage a wide range of domotic devices (sometimes, even of different technologies). The residential gateway offers a unified user interface to send commands and receive information about the state of the domotic devices. This user interface is neither friendly nor intuitive. The aim of the project presented here is to synchronize and represent the residential gateway information in the virtual model representation of the building.
- **Information Exchange Module.** Software module that gathers and processes the information available in the residential gateway (configuration and status of devices). It converts this information into a format compatible with the Visualization and Development Tool. Reversely, it translates any command provided by the VR user and sends it to the residential gateway.
- **Main application module.** Software module that virtually recreates the real environment. It also generates the virtual interface for the user, providing menus for interaction and representing in a completely visual scenario the updated status of devices, rooms, etc.
- **Visualization interface.** It shows the virtual scene in both non-immersive systems (PC with desktop application and simple peripherals) and in immersive systems (with partial or total immersion).

The VDS is managed by a control module, responsible for the control and synchronization of the whole system. As shown in Fig.3, the use case analysis can be broken down into the following actors, use cases and dependencies among them:

- **Actors:**

- **Residential gateway:** system that communicates with the domotic devices, using a domotic communication protocol.
- **Navigation interface:** manages the user's navigation within the virtual model.
- **Initialization and synchronization routine:** establishes the connection with the residential gateway, thus guaranteeing the correspondence between the virtual model and the real environment from the start.
- **Network module:** establishes and keeps a bidirectional communication with the residential gateway.
- **Event listener server:** stays permanently registered to the residential gateway waiting for the reception of events. It notifies those events to the control module, as soon as they happen.

- **Use cases:**

- **Receive events:** the residential gateway notifies to the control module when a change of status takes place (change in a sensor's value, a device is activated, etc.).
- **Send actions:** the user sends actions (commands) from the navigation interface of the virtual model or the user physically activates a device.
- **Routine execution:** the model contains pre-defined scenes (executed as a set of actions).
- **Model configuration:** when initialization takes place, the initial distribution of the registered locations and devices is downloaded from the residential gateway. It is also periodically activated (at the request of the synchronization routine).

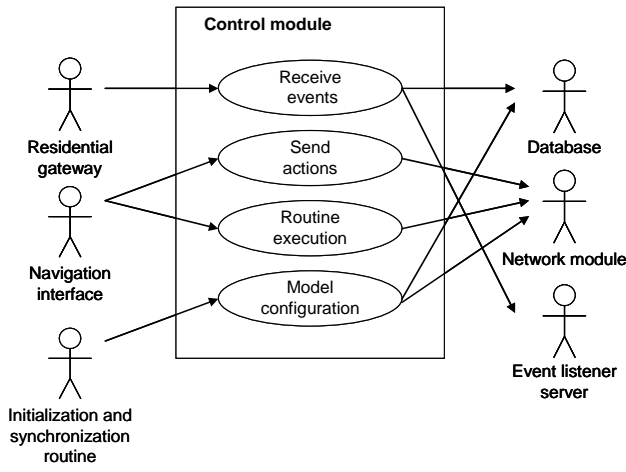


Fig. 3. Use case diagram of the control module

There are two main blocks in the general architecture responsible to make the system communicate with the domotic network and behave accordingly. These are the Information Exchange Module and the Main Application Module presented with more detail in the following sections.

III. INFORMATION EXCHANGE MODULE

The Information Exchange Module (IEM) block is responsible for establishing the communication between the residential gateway and the VDS. The main goal of the IEM is to achieve the synchronization of status and the transmission of events from the residential gateway to the Virtual Reality environment and vice-versa.

The module is based on the sequential exchange of XML messages [9], thus eliminating the need to store the information on the devices' status in intermediate components (such as databases). The use of XML is not new in domotics as it has previously been used to provide a semantic layer on which novel value-added services can easily be built [10]. This concept has made possible the communication between the VR environment and the set of domotic appliances.

Consequently, the Information Exchange Module has two main blocks:

- The connection and configuration module. It analyzes and builds the XML messages.
- The event listener server. This server must be registered at the residential gateway. To do so, the server sends a message to the residential gateway with its IP address and its communication port. The residential gateway is programmed to send its messages to all entities registered in it. So, once registered, the event listener server receives any incoming message from the residential gateway in real time. Every time a device status changes, the residential gateways sends a message each to its registered entities. This is the way to achieve multi-user functionalities. Several VDS applications can be registered in the residential gateway, in such a way that all them can communicate and operate with it.

One of the main challenges of this project is the integration of the network functionalities necessary for the development of this module. This has been achieved by programming separate functions using Visual C++ that are later used by the Visualization and Development Tool as described in section III.2 of this document.

III.1. Connection and configuration module

The main task of the connection and configuration module is to perform the parsing of the XML message, according to the semantic rules of the Document Type Definition (DTD). With this working principle we guarantee the scalability and adaptability of the overall system [9].

The connection and configuration module downloads the information on the locations and devices registered in the residential gateway from the real world. This process is carried out in the boot of the system in order to initialize the virtual model according to the real status of the devices, but also during runtime, thus ensuring permanent synchronization between the domotic devices and their virtual representation. With this process, changes in devices in the real world (for instance, a light is switched on) are immediately displayed on the VDS.

DTD defines the following fields to be included in the XML messages:

- Locations: name of places (rooms, areas) registered in the residential gateway and devices installed in each location.
- Devices: types of domotic devices and their current status.
- Action: description of the event that can be executed by a specific device.
- Routines: set of actions involving one or more devices (also called scenes).

As a result of the parsing process, the information obtained is stored in a set of tables (see description in Table 1). These tables are built dynamically when the connection between the residential gateway and the Virtual Domotic System is established. Moreover, they are asynchronously accessed by the control module and the visualization interface.

Table 1. Set of tables obtained after the parsing process

| NAME | DESCRIPTION |
|-------------------|--|
| Locations table | Stores the locations registered in the residential gateway at the moment of connection. Contains the list of the devices associated to them. |
| Devices table | Contains the list of devices working in each location, together with their id, category and type. |
| Device type table | Stores detailed information about the actions possible for each type of device (switches, dimmers, shutters, thermostats, etc.). |
| Scenes table | Consists of a set of scenes (or routines) that the residential gateway can translate in a given subset of device actions. |

As soon as the system has read the XML messages and built the tables, it is ready to start its two main activities:

- Invoke actions and scenes. When the user selects a specific action or scene from the virtual interface, the order is translated into a given URL address and is sent to the residential gateway via HTTP.
- Receive events. The module interprets the events received from the residential gateway by the event listener server and modifies the configuration tables accordingly.

The aforementioned modules have been implemented using the visualization and development tool “Virtools” [11]. This tool is based on a graphic interface, where the execution line is established through the interconnection of basic behavior functions called Building Blocks (BBs). These BBs can be grouped into Behavior Graphs and scripts in order to achieve more complex results.

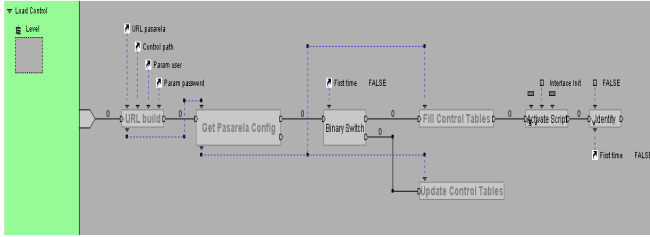


Fig. 4. Example of implementation. Building Blocks of the Control Routine

III.2. Event listener server

This module receives events related to the change of status of the devices. “Event” is understood as a message sent by the residential gateway with data describing the new status of a particular device. The server allows the system to be synchronized with the residential gateway as long as all messages notifying changes in status are received. As this hypothesis can only be assumed in an ideal environment, the complete synchronization is achieved by periodically polling the current situation to the residential gateway as described in section 3.1 for the boot process.

The residential gateway implements a service to register “listeners” interested in receiving status messages whenever a change happens. Messages are then sent to all interested users allowing the system to work properly in a multi-user environment. Fig. 5 shows the structure of the Event Listener Server, comprising the following main blocks:

1. Get events: implements the event listener and the parsing of information contained in the XML message.
2. Device finder: checks that the event refers to a device already stored in the tables. If the device is not found, then the message is considered as “not valid” and is ignored by the server.
3. Get parameters: processes the data in the messages, obtaining the values to be updated in the tables.
4. Update database: stores the new values in the tables and activates routines to modify the state of the devices in the virtual model.
5. Send notice: sends a message to the user interface to notify changes of status.

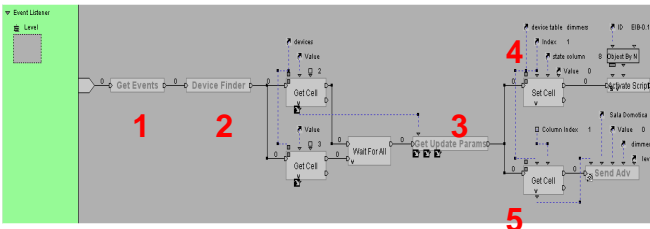


Fig. 5. Script of the Event Listener Server

IV. MAIN APPLICATION MODULE

The Main Application Module has three significant features:

- recreates the rooms and the distribution of the domotic devices in the real environment virtually,
- presents the information on the real status of the devices in a visual and intuitive way,
- provides interaction menus to configure the real environment.

IV.1. 3D modeling of home/building environment

The 3D model of the real home or building has been created, in this particular project, using 3DStudio Max 9 (Autodesk), a tool capable of designing any type of object that also provides files formats suitable to be imported with Virtools. 3DStudio translates .MAX files into .NMO format, compatible with Virtools.

The environment has been built starting from the CAD files of the building, containing detailed information about dimensions, position of walls, pillars, etc., of the building.

The next step was the recreation of objects’ textures and materials, paying special attention to their illumination. Firstly, light processing was developed using Maxwell Render, a powerful tool for illumination, but finally the implementation of textures and light was carried out by means of lightmaps. Lightmaps are textures that automatically save the materials and light of objects. They reduce the load of processing during runtime, although they present a static shading of objects (invariant position of the light in the model).



Fig. 6. 3D model of CeDInt-UPM building

IV.2 Navigation tools

The navigation system developed in the project allows total interaction with the virtual model by using different hardware inputs.. According to the preferences of the user, the navigation can be accomplished through a mouse and keyboard or a joystick/PAD. Two navigation modules have been developed:

- Cursor navigation: declares the keys that can be used to navigate, as well as the speed of the movement and changes in direction (also implemented with W, X, A and D keys).
- Master navigation: controls the collision detection with obstacles. This is an important factor when realistic navigation is desired, since the user should not be able to go through walls or closed doors in the virtual environment.

IV.3 Interface menu development

The interface menu provides a wide variety of features to make navigation easy and intuitive. These features are presented here through the description of the different scripts the module is made of:

- *Pointer script*: It shows a pointer in the centre of the screen. When the user moves around the scene and the pointer targets a domotic device, this specific device is highlighted in red and a sound is emitted. In this way, the user easily knows which objects are capable of being activated by the residential gateway.
- When the user selects a device, the menu of specific actions available for that device drops down and the user can choose an action to be launched. This sequence of events is depicted in Fig. 7.



Fig. 7. Pointer script features: selection and activation of devices. A) Pointer in the center of the screen. B) Selection of illumination systems (highlighted in red). C) and D) Specific actions menu for illumination devices

- *Interface init*. It composes the menu for any device in the room where the user is. In this way, the system is always ready for any sudden action the user may do.
- *Update menu*. It creates and deletes the buttons with the different actions possible for the selected device at every moment. It creates the buttons but does not show them on the screen.
- *Update scene menu*. It builds the buttons related to the scenes already pre-configured in the residential gateway (they cannot vary during runtime). For example, a scene called "Good Morning" can be configured to raise all shutters and turn off all lights in the building.
- *Parent sliding 2D frame script*. It contains all the action buttons of the selected device and shows/ hides these buttons on/from the screen. The *Parent Sliding 2D Scene Frame script* performs the same function, associated to scenes.
- *Next and Prev scripts*. They allow moving forward and backwards within the list of devices in a room, providing an extra navigation tool.
- *Current room script*. It shows a label with the name of the room where the user is.
- *Event advertisement script*. It creates labels that show modifications in the state of the devices. This is especially important when the system is used in a multi-user environment, since it shows messages about changes the other users have made in other rooms. Moreover, another user could physically activate or adjust a device and the system will notify it to the VR user by means of these labels.
- *Temp Display Script*. In case a room has a thermostat-type device, the system shows the configuration temperature of the heating/air conditioning system in the room.

V. THE SYSTEM UP AND RUNNING

In the previous sections, the general structure of the system, its main building blocks and working principles have been presented. This section describes the specific characteristics of the Virtual Domotic System that has been successfully implemented and is currently up and running.

In order to implement a real system, the following technical solutions have been chosen, regarding the different elements of the architecture:

A set of EIB (European Installation Bus) devices have been installed and configured through the ETS3 tool [12]: blind controllers, switchers, dimmers, air conditioning control and RS-232 interfaces. These devices have been connected to an OSGi-based platform [13], [14], from eNEO Labs.

The virtual environment modeled in the project corresponds to the domotic show-room, the corridor and two offices located on the first floor of the Centre for Integral Domotics of the UPM. As visualization infrastructures, an immersive stereowall and non-immersive PC environment have been tested.



Fig. 8. System set-up: stereowall, domotic devices installed on a show-panel and residential gateway.

The following figures show the system working both in the real building and its virtual model.

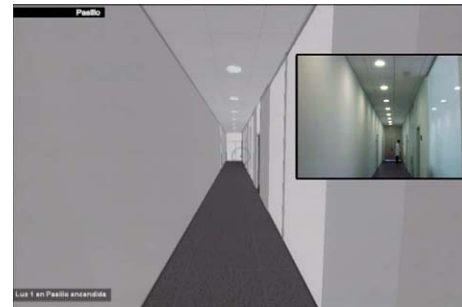


Fig. 9. Activation of lights in the corridor

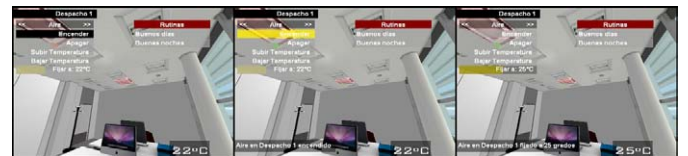


Fig. 10. Office 1. Selection or heating system. Activation. Modification of temperature.

ACKNOWLEDGMENTS

Authors would like to thank the Spanish Ministry of Industry, Tourism and Commerce, for the support provided through the funded projects FIT-350300-2007-41 and TSI-020100-2008-402.

REFERENCES

- [1] S. Tompro, N. Mouratidis, M. Caragiozidis, H. Hrasnica, A. Gavras. *A Pervasive Network Architecture Featuring Intelligent Energy Management of Households*. 1st International Conference on Pervasive Technologies Related to Assistive Environments. Athens, Greece. 2008.
- [2] D. Bonino, E. Castellina, F. Corno, A. Garbo. *Control Application for Smart House through Gaze interaction*. COGAIN 2006. Turin Italy. 2006.
- [3] D. Bonino, E.Castellina, F.Corno, A. Gale, A. Garbo, K. Purdy, F.Shi. *A Blueprint for Integrated Eye-Controlled Enviroments*. Universal Access in the Information Society. Vol 8/4. ISSN: 1615-5289. 2009.
- [4] E. Haik, T. Barker, J. Sapsford, S. Trinis. *Investigation into Effective Navigation in Desktop Virtual Interfaces*. Proceedings of the seventh international conference on 3D Web technology. Tempe, Arizona. 2002.
- [5] A.I Molina, M.A. Redondo & M. Ortega: *Virtual Reality for Teaching Domotics*. IADIS International Conference Applied Computing. Lisbon, Portugal. 2004.
- [6] F. P. Brooks. *What's Real About Virtual Reality?*. IEEE Computer Graphics And Applications, 19(6), 16. 1999.
- [7] R. S. Kalawsky. *The Science of Virtual Reality and Virtual Environments: A Technical, Scientific and Engineering Reference on Virtual Environments*. Addison-Wesley, Wokingham, England . 1993.
- [8] G. Burdea, P. Coiffet. *Virtual Reality Technology*. JohnWiley. New York. 1994.
- [9] E. R. Harold. *XML 1.1 Bible*. 3rd Edition. John Wiley & Sons, Inc. London. 2004.
- [10] F. Furfari, L. Sommaruga, C. Soria, R. Fresco. *DomoML: the Definition of a Standard Markup for Interoperability of Human Home Interactions*. Proceedings of the 2nd European Union symposium on Ambient intelligence. ISBN:1-58113-992-6. Eindhoven. 2004.
- [11] C. Mc Carthy, D. Callele. *Virtools Dev User Guide*. 2004.
- [12] *Instabus EIB Commercial Catalogue*. Siemens, S.A. 2003.
- [13] OSGi Alliance. *OSGi Service-Platform*. Release 2, October 2001. IOS Press. 2002.
- [14] S. Haiges. *OSGi Tutorial. A Step by Step Introduction to OSGi Programming*. October 2004.



Fig. 11. Domotic show-room. Opening of shutters.



Fig.12. VDS running in a CAVE.

VI. CONCLUSIONS AND FUTURE RESEARCH LINES

This paper presents the conceptual structure of a Virtual Domotic System and its successful implementation in a fully functional real system, thus showing the technical feasibility of the idea.

The principal goal achieved with the VDS has been the communication and synchronization between a residential gateway and an application based in Virtual Reality. The IEM makes this possible using XML to send actions and receive data related to the current state of the devices. This makes the system easy to integrate with any residential gateway with internet connection capabilities.

The VDS provides a novel and enhanced user interface for residential gateways, based on 3D virtual environments and interfacing techniques. As described in the introduction, the usability and attractive interfaces are the main issues for user acceptance of domotic systems so VR can definitely help to resolve this handicap.

This system allows interacting with home appliances modeled in a virtual environment using real domotic devices. In this way, it introduces a new 3D interaction technique where users can change the state of the virtual world through the different appliances installed in their buildings.

Among other features, the Virtual Domotic System is a fully multi-user system, allowing the simultaneous interaction of several users both in the virtual application and physically in the real building. This facilitates the use of residential gateways and control systems in new applications such as maintenance and surveillance of big buildings.

The VDS opens the possibility to develop and integrate additional features, such as a room editor to easily create 3D environments, inclusion of video streaming devices (for information and surveillance purposes), support of different types of sensors (water/gas escapes, user presence, opened doors/windows, etc.), and the generalization of the system for any type of residential gateway.